

추가 문제

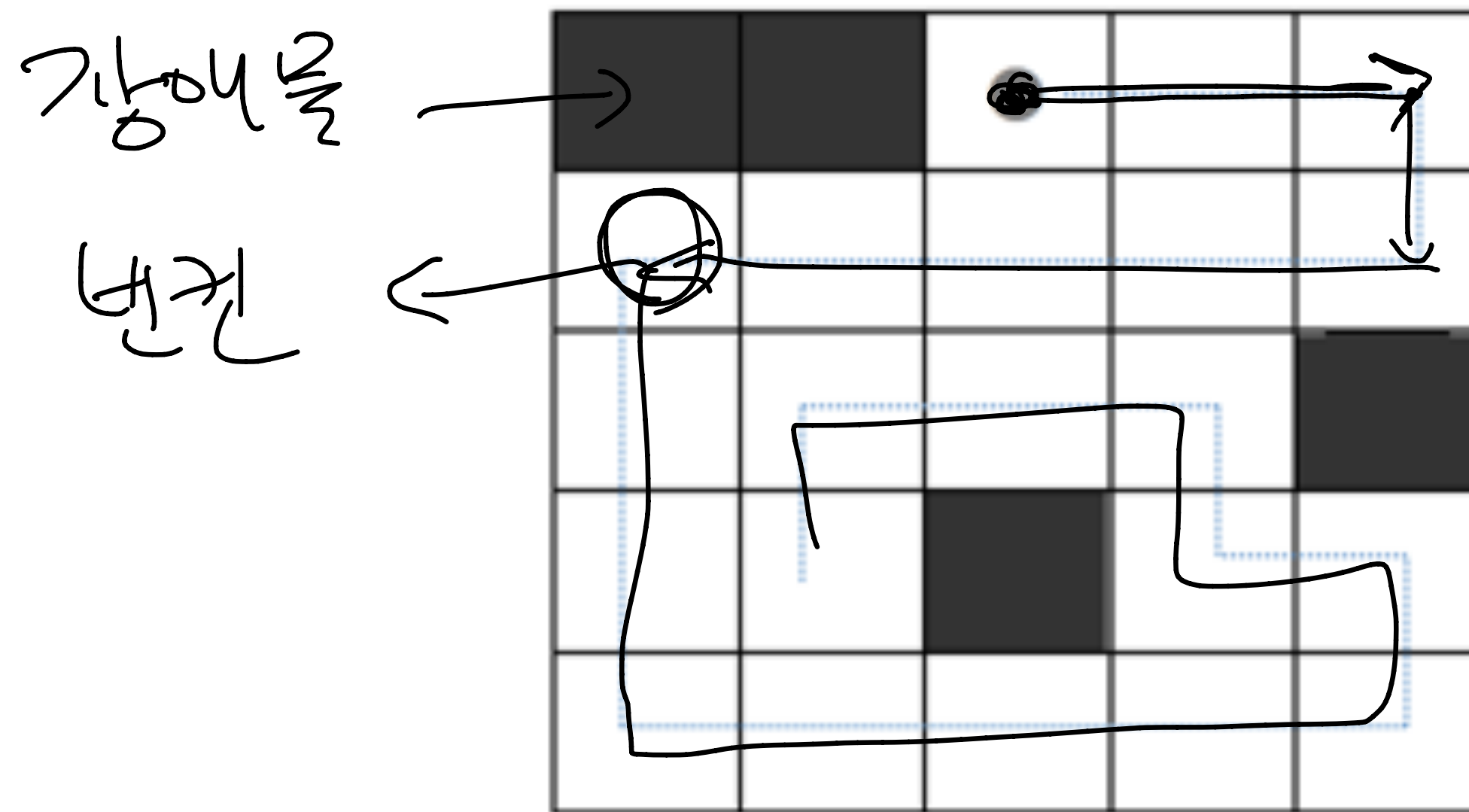
최백준 choi@startlink.io

NxM 보드 완주하기

<https://www.acmicpc.net/problem/9944>

- NxM 보드가 있다. 보드의 각 칸은 빈 칸 또는 장애물이다.
- 보드 위에 공을 하나 놓고, 모든 칸을 방문하려고 한다.
- 공은 네 방향 중 한 방향으로 더 이상 이동하지 않을 때까지 이동한다.

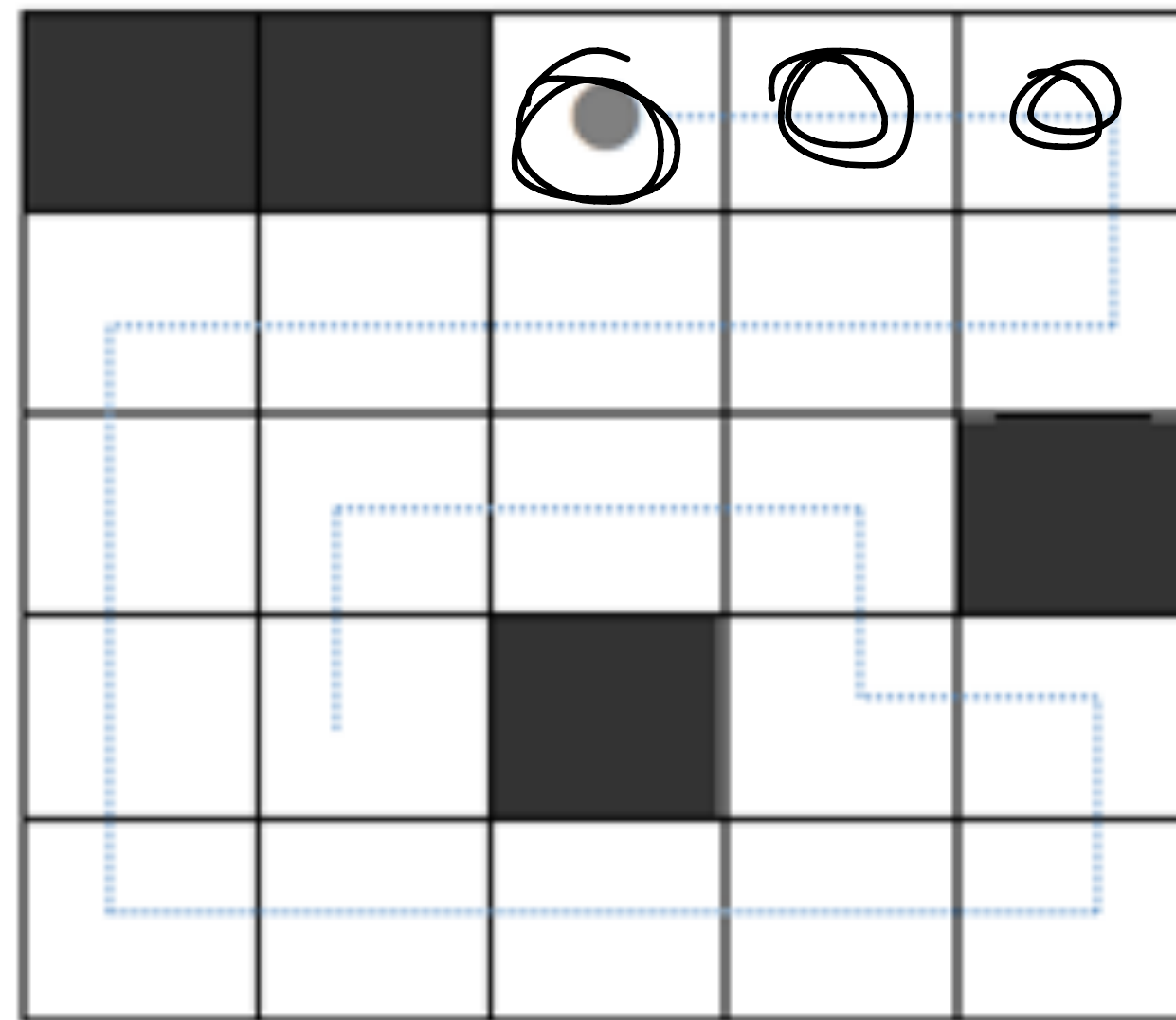
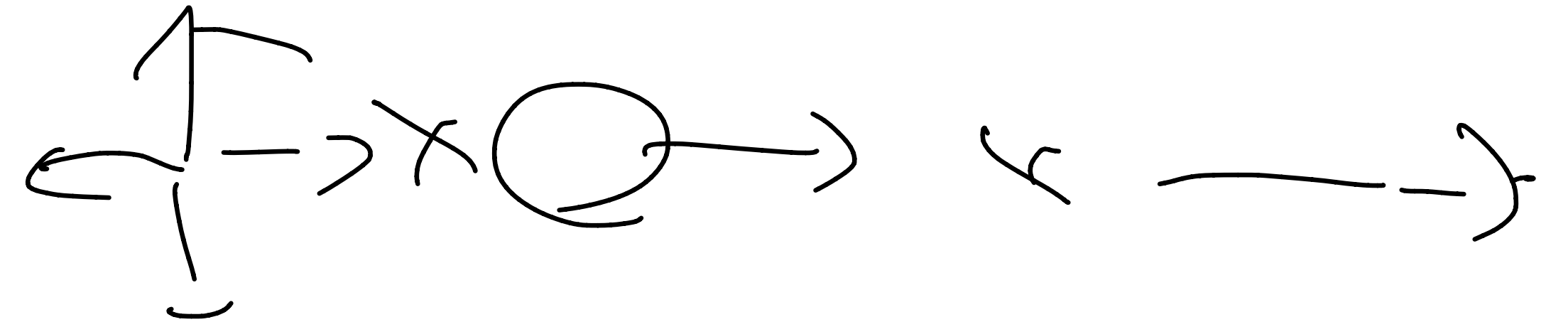
최소 이동



NxM 보드 완주하기

<https://www.acmicpc.net/problem/9944>

- 보드의 모든 칸 위에 공을 놓았다고 가정하고
- 모든 네 방향을 다 이동해본다.



NxM 보드 완주하기

<https://www.acmicpc.net/problem/9944>

- 소스: <http://codeplus.codes/ce26e37f6cdb46b5b209324d4b892272>

수족관 1

<https://www.acmicpc.net/problem/8982>

선분: $N \leq 5,000$
좌표 $\leq 40,000$

5

- 수족관에 물이 가득 차 있고, 구멍을 하나 뚫었을 때 물이 빠진다.
- 물의 남은 양을 구하는 문제

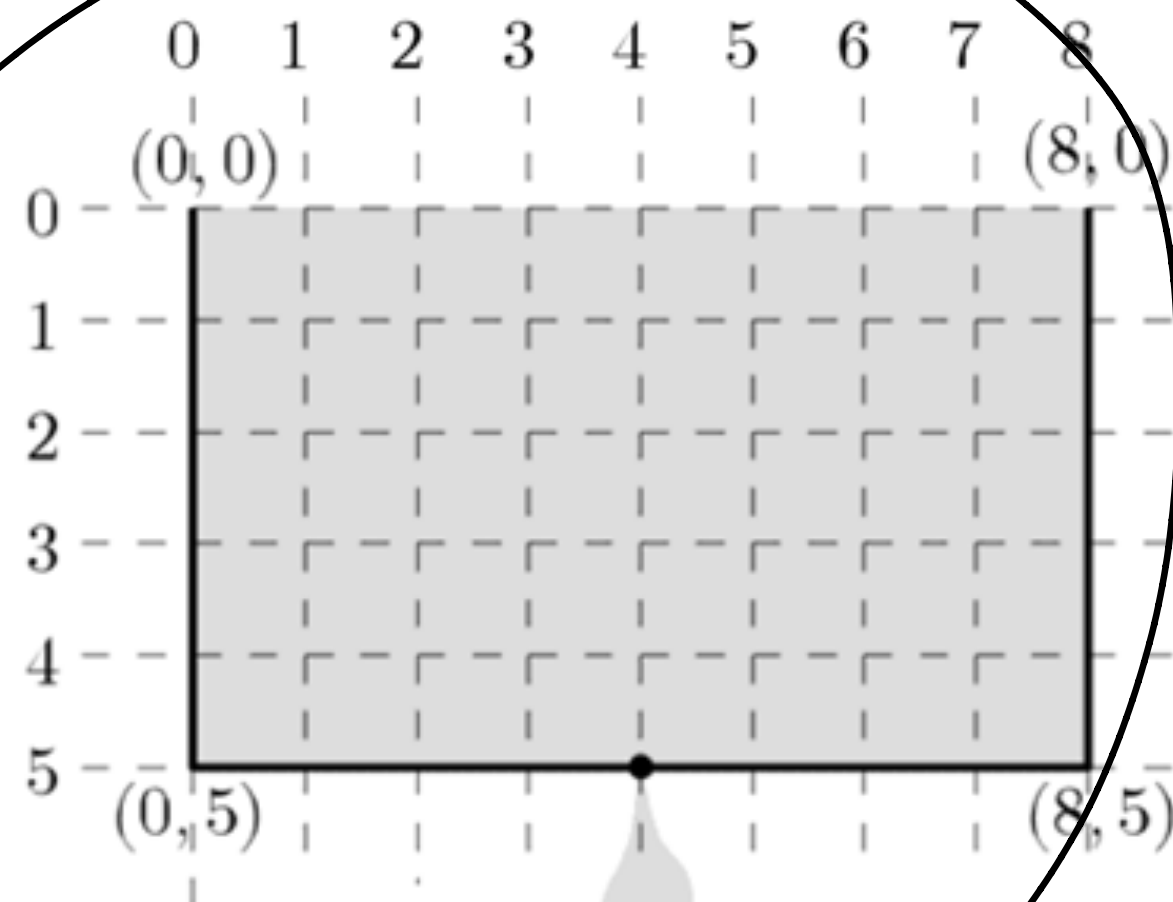


그림 1. 수족관과 구멍.

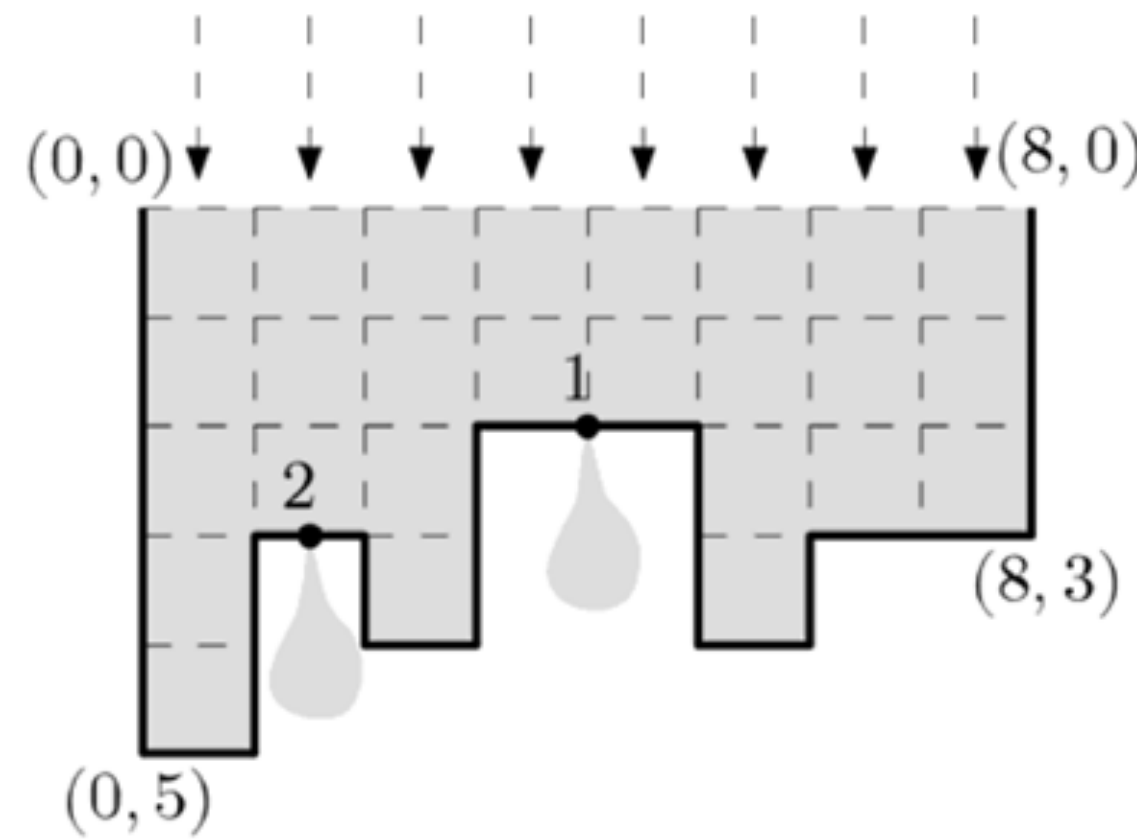


그림 2. 가득 찬 물의 양은 26L, 구멍은 2개.

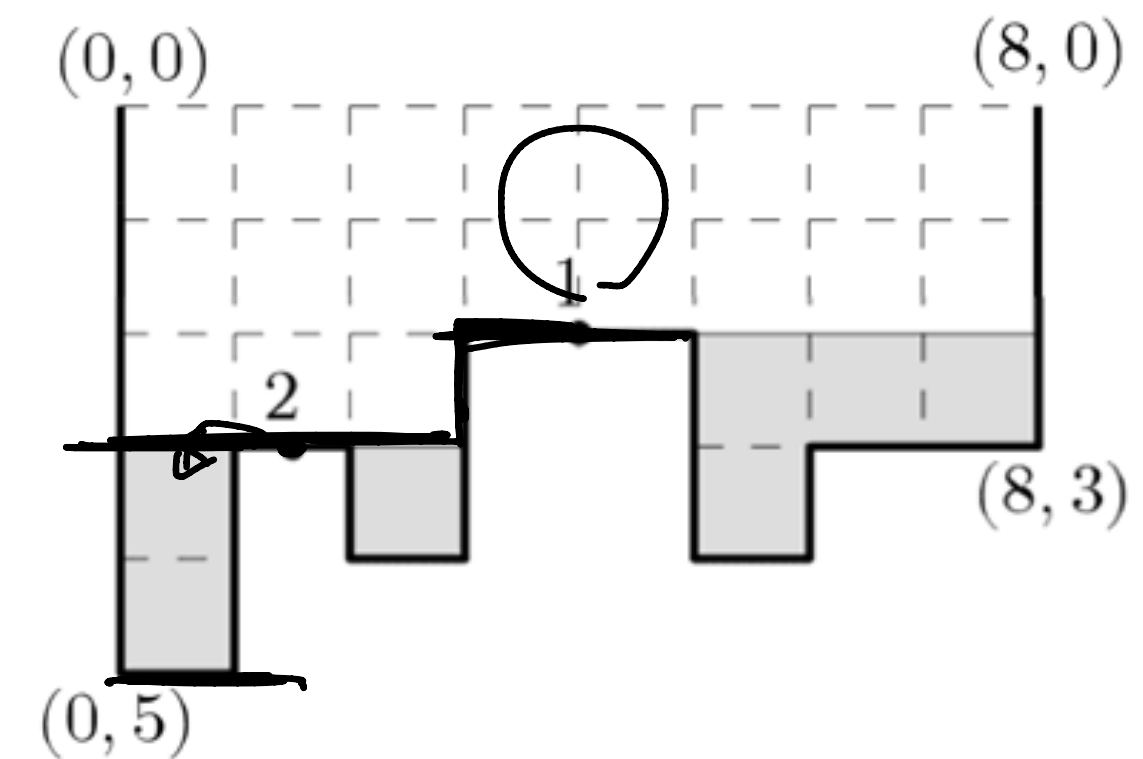


그림 3. 마지막에 남은 물의 양은 7L.

수족관 1

6

<https://www.acmicpc.net/problem/8982>

- 두 개의 배열을 이용했다.
- $\text{bottom}[x]$ = x좌표가 x 인 곳의 바닥 y좌표
- $\text{top}[x]$ = x좌표가 x 인 곳이 물이 얼마나 찰 수 있는지

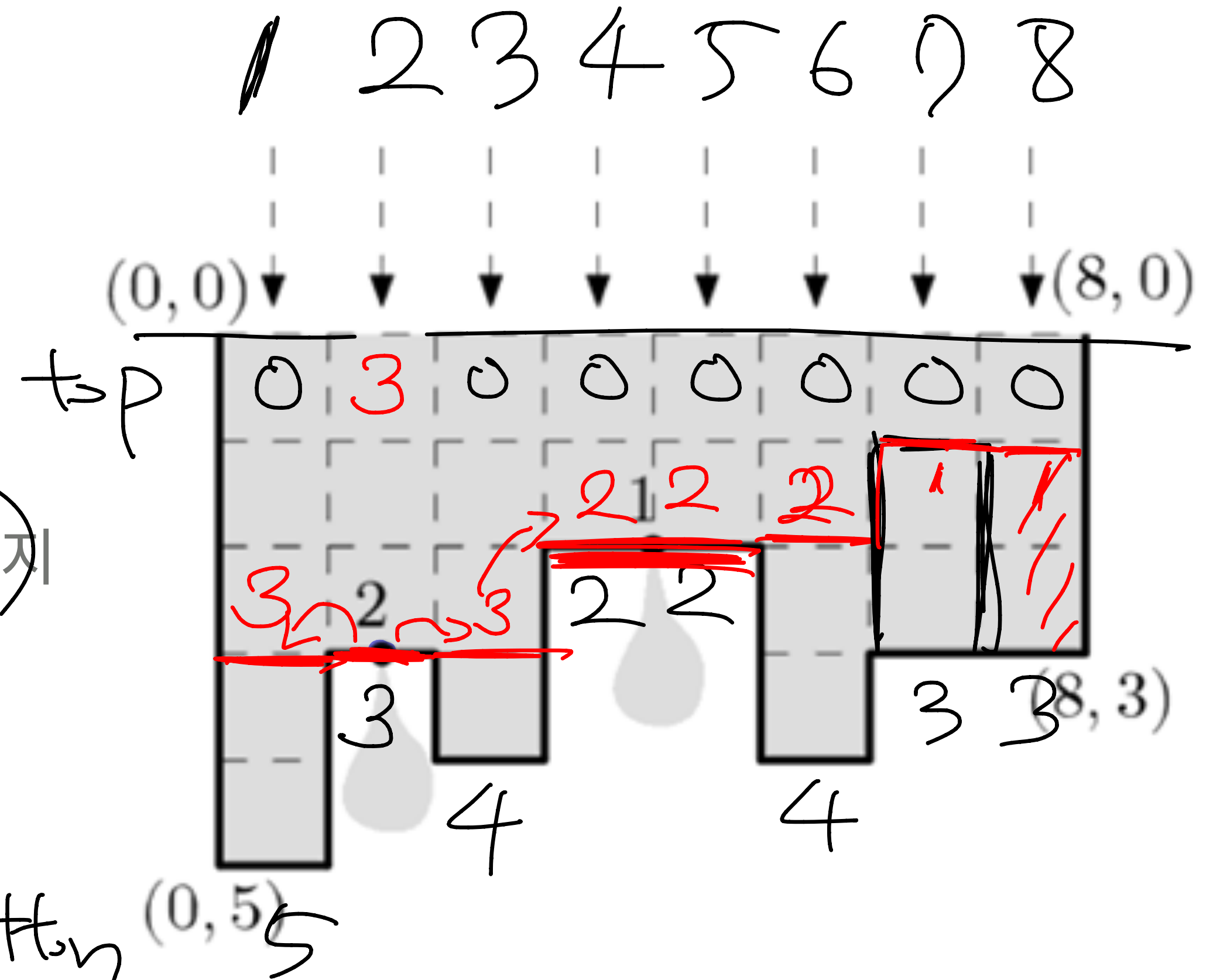


그림 2. 가득 찬 물의 양은 26L, 구멍은 2개.



수족관 1

<https://www.acmicpc.net/problem/8982>

- 소스: <http://codeplus.codes/0ea25adaec81448a8edb9f6938e84fac>
- 이 방법은 40,000² 만큼의 시간이 필요해 너무 오래 걸린다.

- 같은 선분에 대해서 여러 번 처리를 할 필요가 없다.
- 선분 단위로 그룹을 지어서 처리를 하면 N^2 으로 줄일 수 있다.

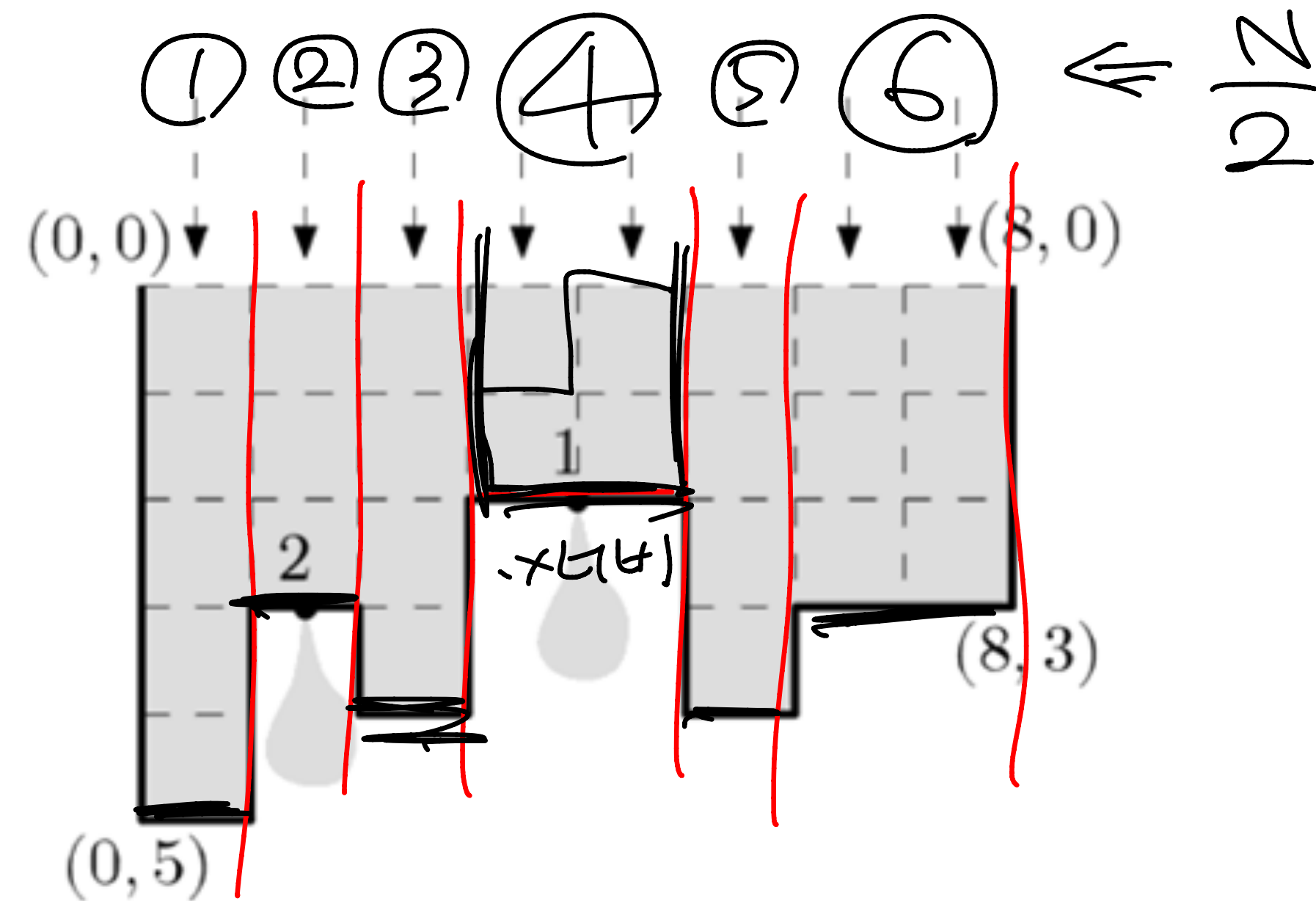
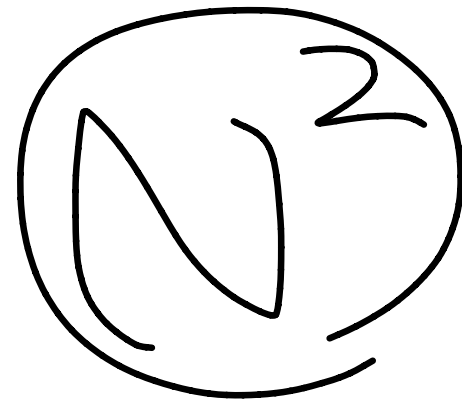


그림 2. 가득 찬 물의 양은 26L, 구멍은 2개.

수족관 1

<https://www.acmicpc.net/problem/8982>

- 소스: <http://codeplus.codes/736f264f7df44afcb4692c3404db6abd>

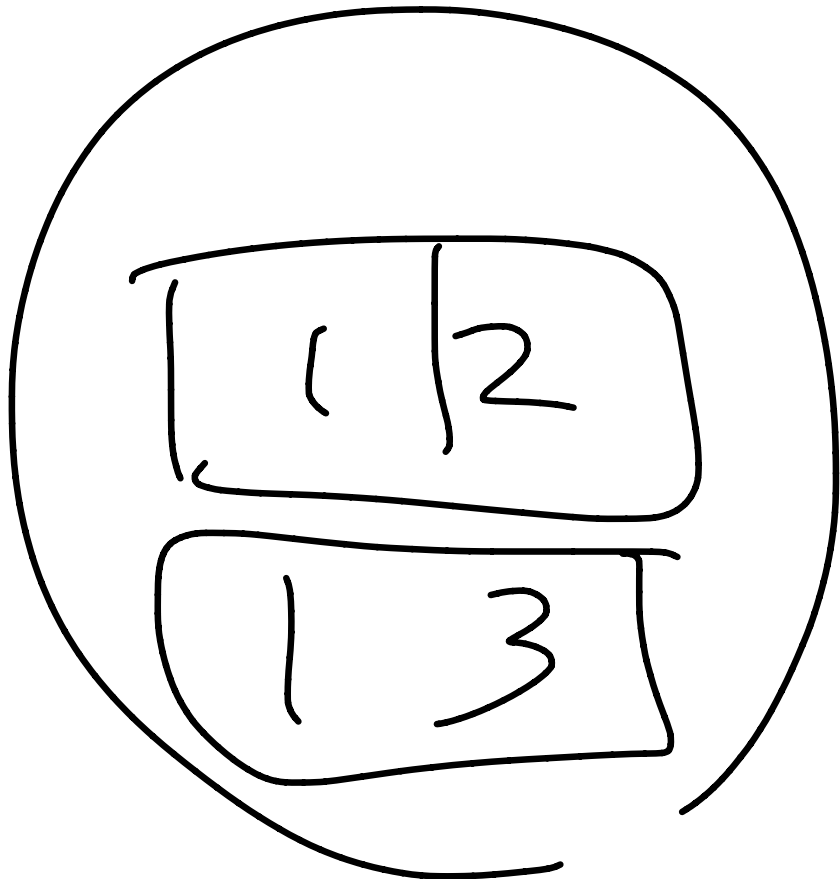


스도미노쿠

<https://www.acmicpc.net/problem/4574>

- 스도미노쿠를 푸는 문제

1 2
3 4 5
6 7 8



$$\frac{9 \times 8}{2} = 36$$

	1	2	3	4	5	6	7	8	9
A	7	2	3	4					5
B		6	1				8	4	2
C			9	2	1	8			7
D			6						3
E	7		8		6				
F	4			3	2		7	6	
G				5	9				
H									
I	9			4				7	8

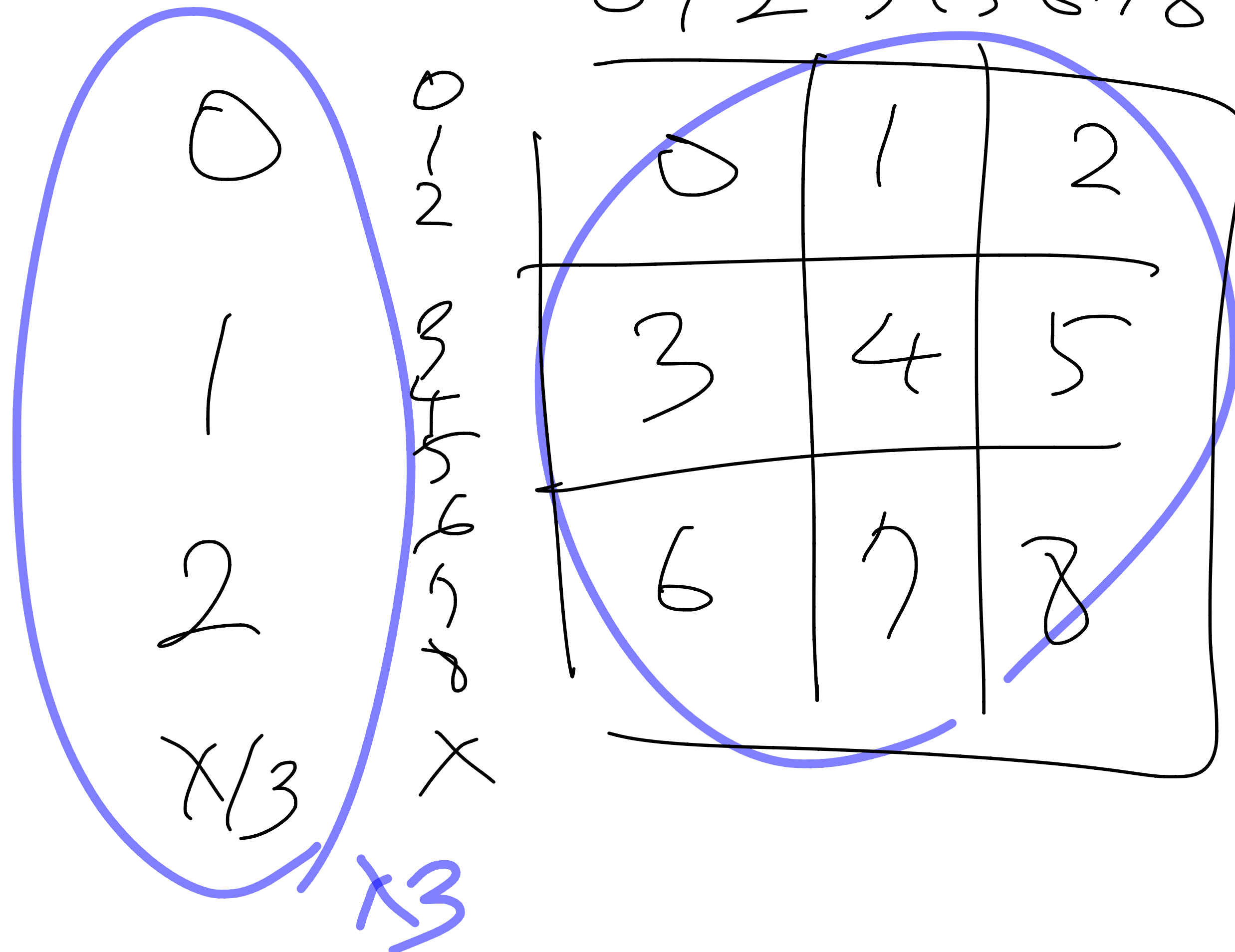
	1	2	3	4	5	6	7	8	9
A	8	7	2	6	4	3	1	9	5
B	3	6	1	9	7	5	8	4	2
C	5	4	9	2	1	8	6	3	7
D	1	2	6	7	5	4	9	8	3
E	7	3	8	1	6	9	2	5	4
F	4	9	5	8	3	2	7	6	1
G	2	8	4	5	9	7	3	1	6
H	6	5	7	3	8	1	4	2	9
I	9	1	3	4	2	6	5	7	8

스도미노쿠

<https://www.acmicpc.net/problem/4574>

11

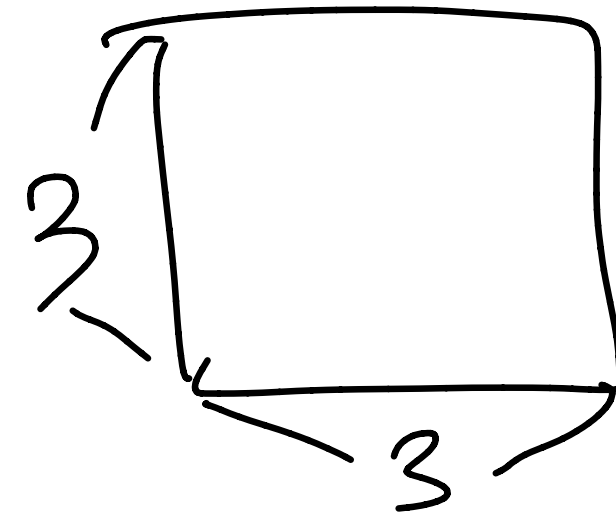
- 첫 번째 칸부터 수를 하나씩 차례대로 채워본다.



$Y/3$
y

스도미노쿠

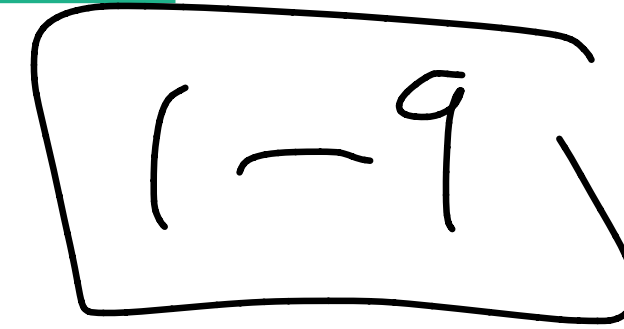
<https://www.acmicpc.net/problem/4574>



12

- 소스: <http://codeplus.codes/d48e5965a67e4f6b895e767dfdd98793>

행, 열, 3456789



욕심쟁이 판다

<https://www.acmicpc.net/problem/1937>

13

LIS

- $N \times N$ 크기의 이차원 배열에서 최대 부분 증가 수열을 구하는 문제

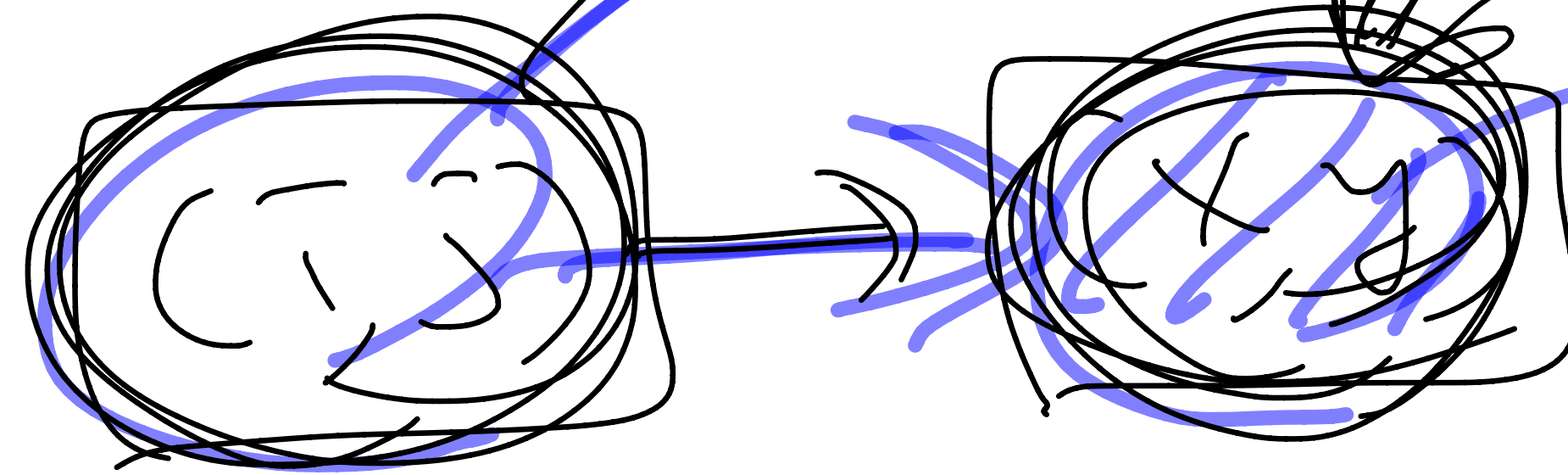
14	9	12	10
1	11	5	4
7	15	2	13
6	3	16	8

욕심쟁이 판다

<https://www.acmicpc.net/problem/1937>

14

- $D[i][j]$ = 판다가 (i, j) 에서 이동을 시작했을 때, 최대한 살 수 있는 일수



$(t, s), (x, y)$ 인 것

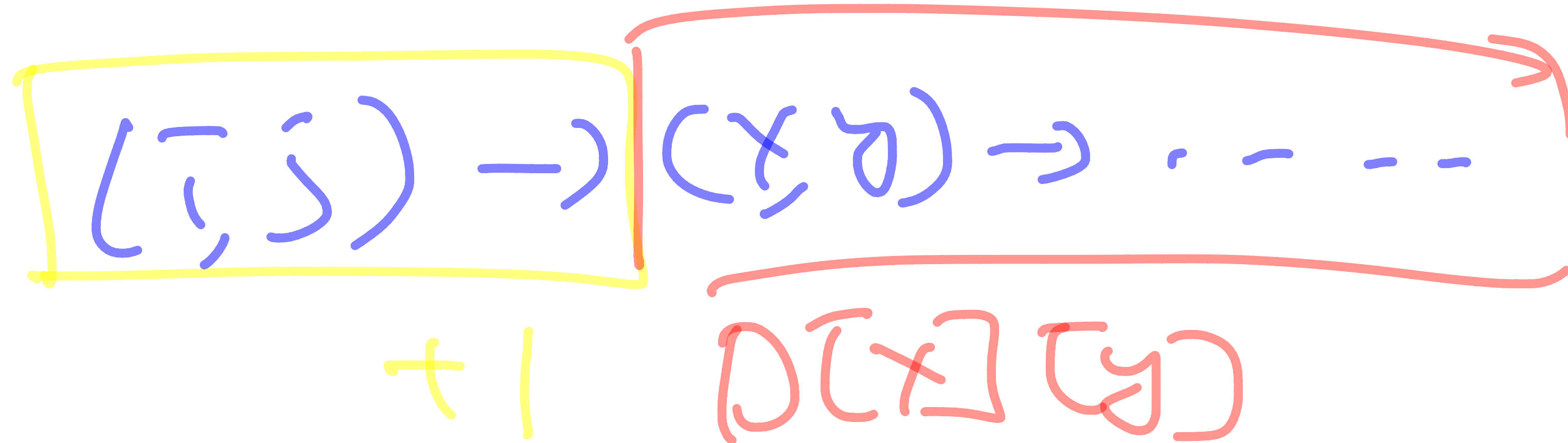
$$A[t][s] < A[x][y]$$

욕심쟁이 판다

15

<https://www.acmicpc.net/problem/1937>

- $D[i][j] = \max(D[x][y]) + 1$
- $(i, j) \rightarrow (x, y)$ 이동 가능
- $A[i][j] < A[x][y]$



욕심쟁이 판다

16

<https://www.acmicpc.net/problem/1937>

- Top-Down 소스: <http://codeplus.codes/a65735cb9033490b8ec994985d9d114e>
- Bottom-Up 소스: <http://codeplus.codes/7ec53c05abed4d1a80e5703e4d0ec2b6>

사탕 게임

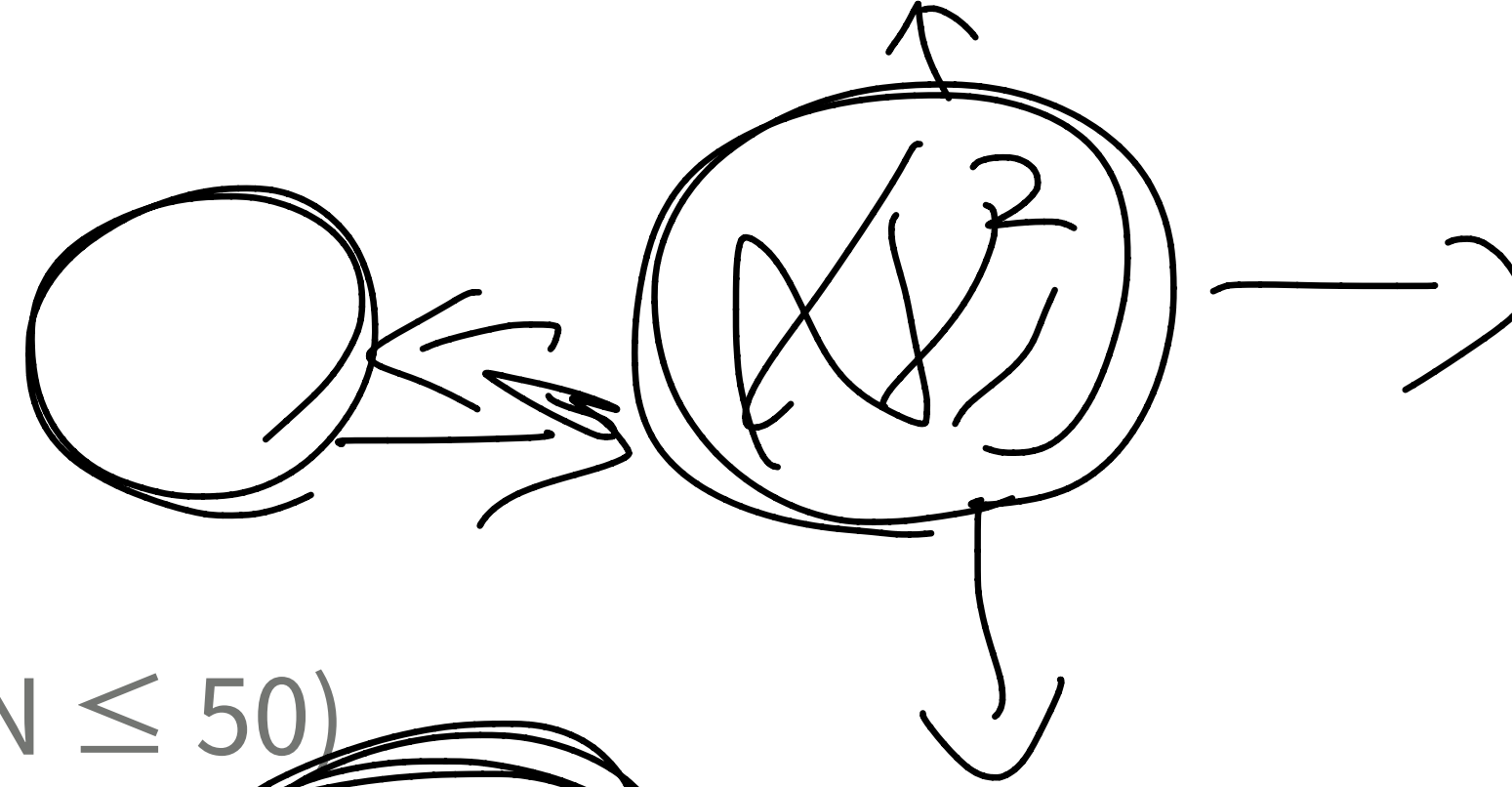
<https://www.acmicpc.net/problem/3085>

- $N \times N$ 크기의 테이블에 사탕이 있다. ($N \leq 50$)
- 인접한 두 칸을 고르고, 사탕을 교환한다.
- 그 다음, 같은 색으로 이루어져 있는 가장 긴 연속 부분 행 또는 열을 고르는 문제

사탕 게임

<https://www.acmicpc.net/problem/3085>

18



- $N \times N$ 크기의 테이블에 사탕이 있다. ($N \leq 50$)

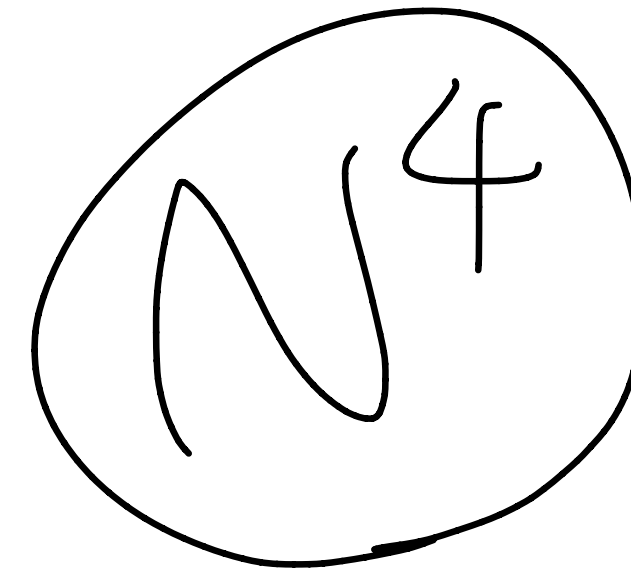
인접한 두 칸을 고르고, 사탕을 교환한다. $\rightarrow (N^2) * 2$ 까지 경우가 가능하다.

- 그 다음, 같은 색으로 이루어져 있는 가장 긴 연속 부분 행 또는 열을 고르는 문제

사탕 게임

<https://www.acmicpc.net/problem/3085>

19



- $N \times N$ 크기의 테이블에 사탕이 있다. ($N \leq 50$)
- 인접한 두 칸을 고르고, 사탕을 교환한다. $\rightarrow (N^2) * 2$ 가지 경우가 가능하다.
- 그 다음, 같은 색으로 이루어져 있는 가장 긴 연속 부분 행 또는 열을 고르는 문제 $\rightarrow O(N^2)$

사탕 게임

20

<https://www.acmicpc.net/problem/3085>

- 소스: <http://codeplus.codes/2b315086a04e4c0d835d758b014426d4>

로봇 조종하기

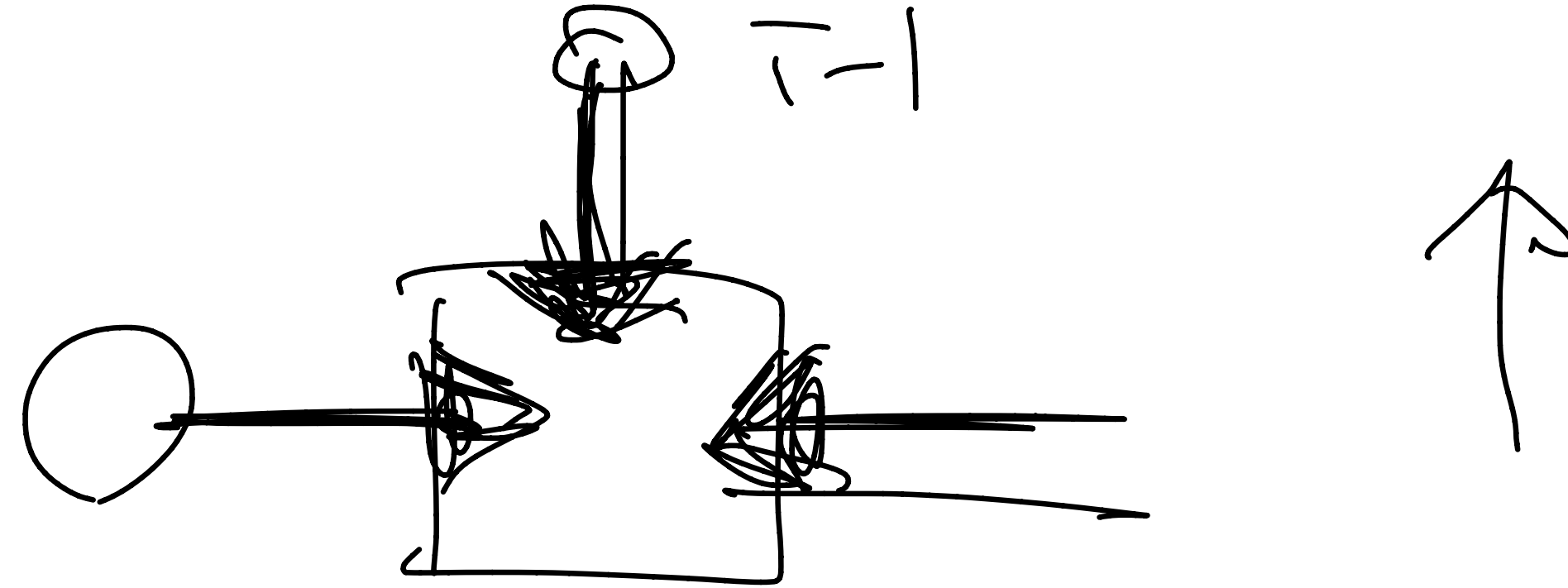
<https://www.acmicpc.net/problem/2169>

- $(1, 1)$ 에서 (N, M) 으로 이동하려고 한다
- 왼쪽, 오른쪽, 아래로만 이동할 수 있다. 합의 최대값을 구하는 문제

로봇 조종하기

<https://www.acmicpc.net/problem/2169>

22

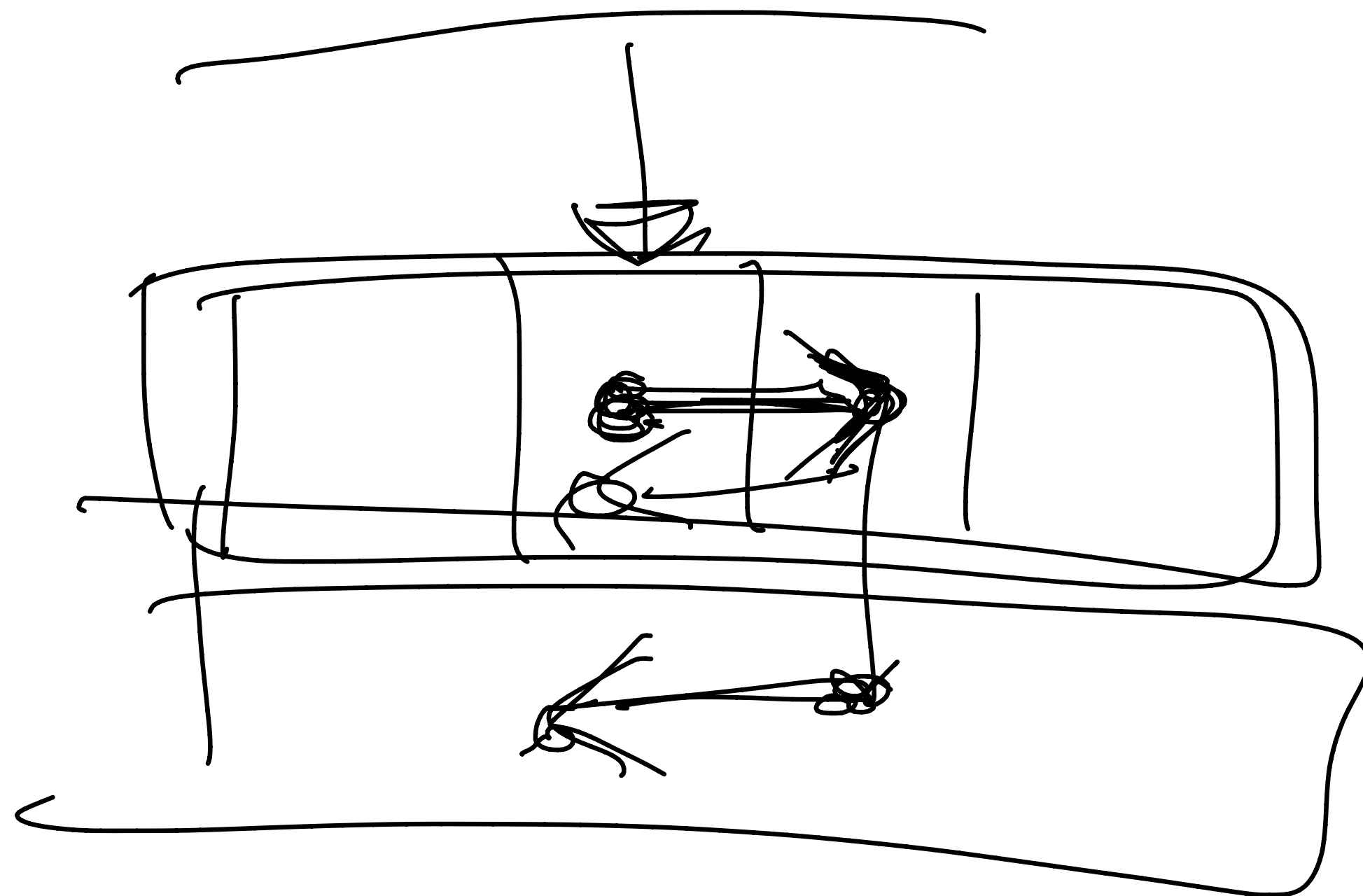


• (1, 1)에서 (N, M)으로 이동하려고 한다

• 오른쪽, 아래로만 이동할 수 있는 문제 = 이동하기

$$D[i][j] = \max(D[i-1][j], D[i][j-1]) + A[i][j]$$

(i, j)



로봇 조종하기

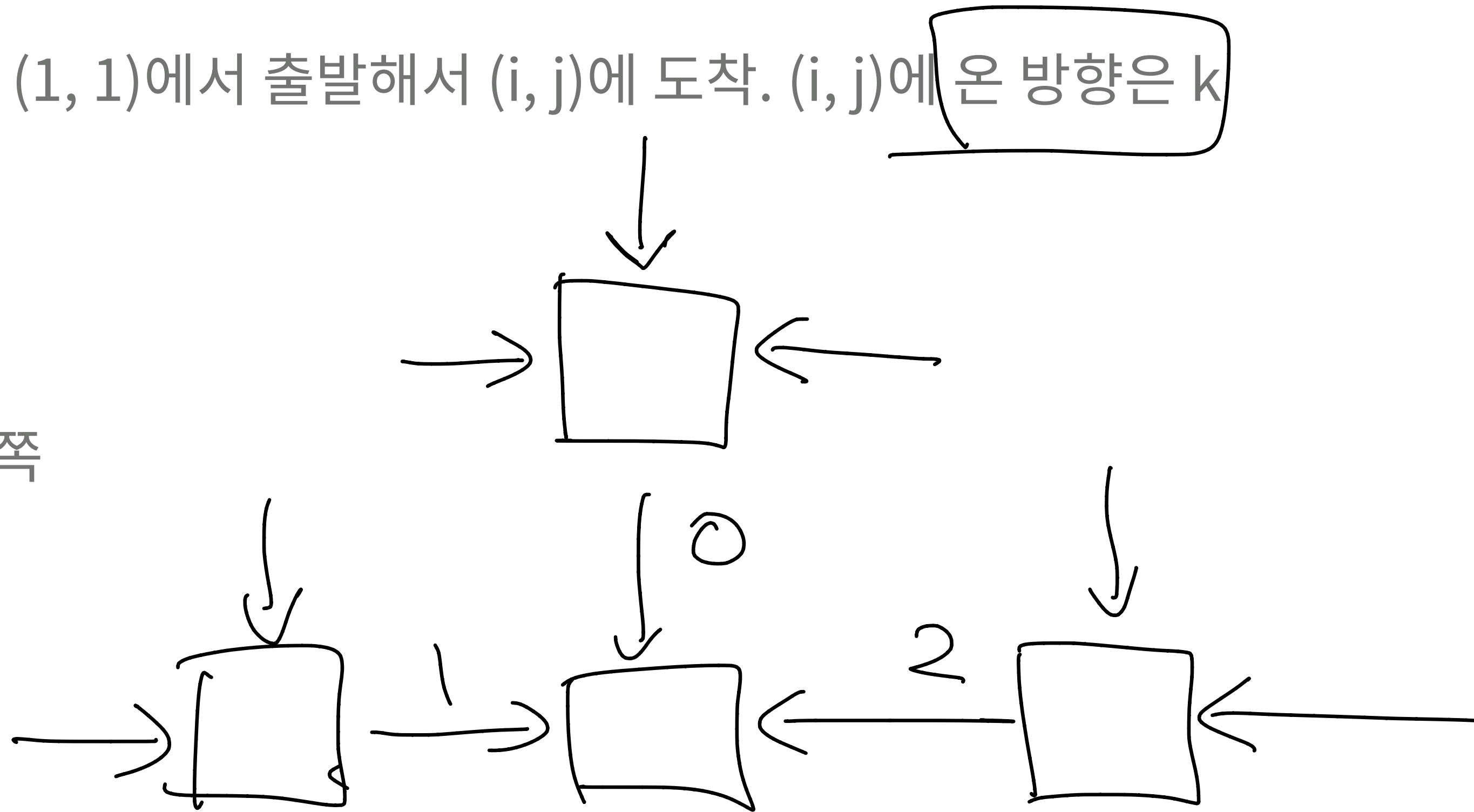
<https://www.acmicpc.net/problem/2169>

- 가장 윗 행부터 차례대로 처리를 하면 된다
- 위, 왼쪽, 오른쪽 순서대로 처리를 해야 한다

로봇 조종하기

<https://www.acmicpc.net/problem/2169>

- $D[i][j][k] = (1, 1)$ 에서 출발해서 (i, j) 에 도착. (i, j) 에 온 방향은 k
- $k = 0$: 위
- $k = 1$: 왼쪽
- $k = 2$: 오른쪽



로봇 조종하기

25

<https://www.acmicpc.net/problem/2169>

• $D[i][j][0]$ =

(0, 1, 2)

• $D[i][j][1]$ =



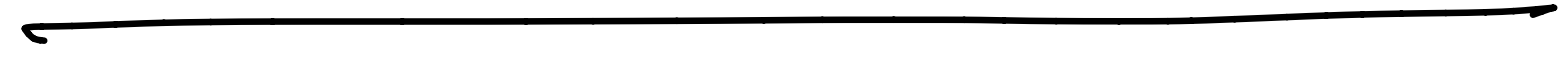
(0, 1)

• $D[i][j][2]$ =

(0, 2)

로봇 조종하기

<https://www.acmicpc.net/problem/2169>

- $D[i][j][0] = \max(D[i-1][j][0], D[i-1][j][1], D[i-1][j][2]) + A[i][j];$

- $D[i][j][1] = \max(D[i][j-1][0], D[i][j-1][1]) + A[i][j];$

- $D[i][j][2] = \max(D[i][j+1][0], D[i][j+1][2]) + A[i][j];$


로봇 조종하기

<https://www.acmicpc.net/problem/2169>

- 소스: <http://codeplus.codes/5390a59a434f4c2e91d1528d39d4ddfa>

가장 큰 정사각형

28

<https://www.acmicpc.net/problem/1915>

- 1로 이루어진 가장 큰 정사각형을 찾는 문제

가장 큰 정사각형

29

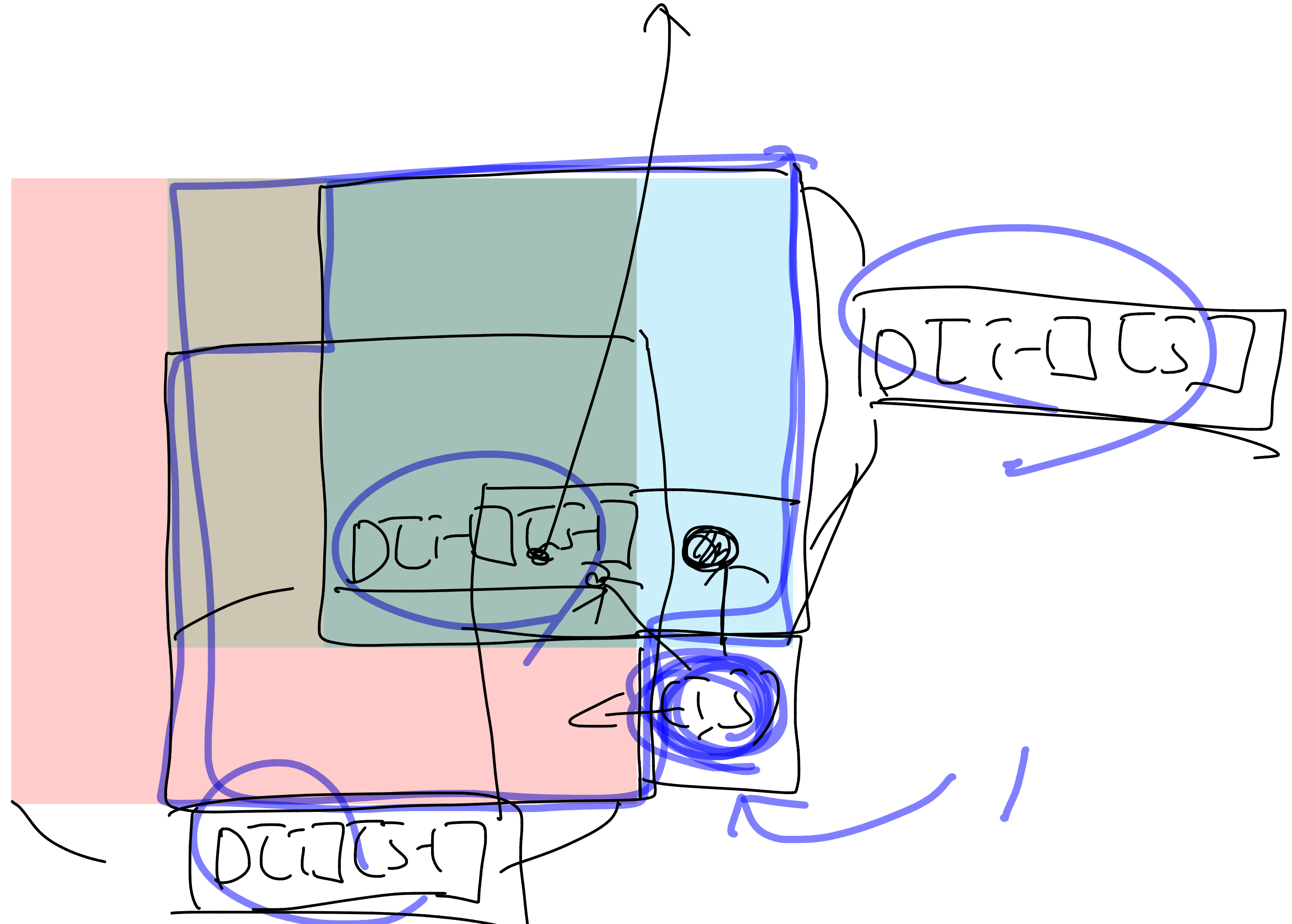
<https://www.acmicpc.net/problem/1915>

• $D[i][j] = (i, j)$ 를 오른쪽 끝으로 하는 가장 큰 정사각형의 변의 길이

가장 큰 정사각형

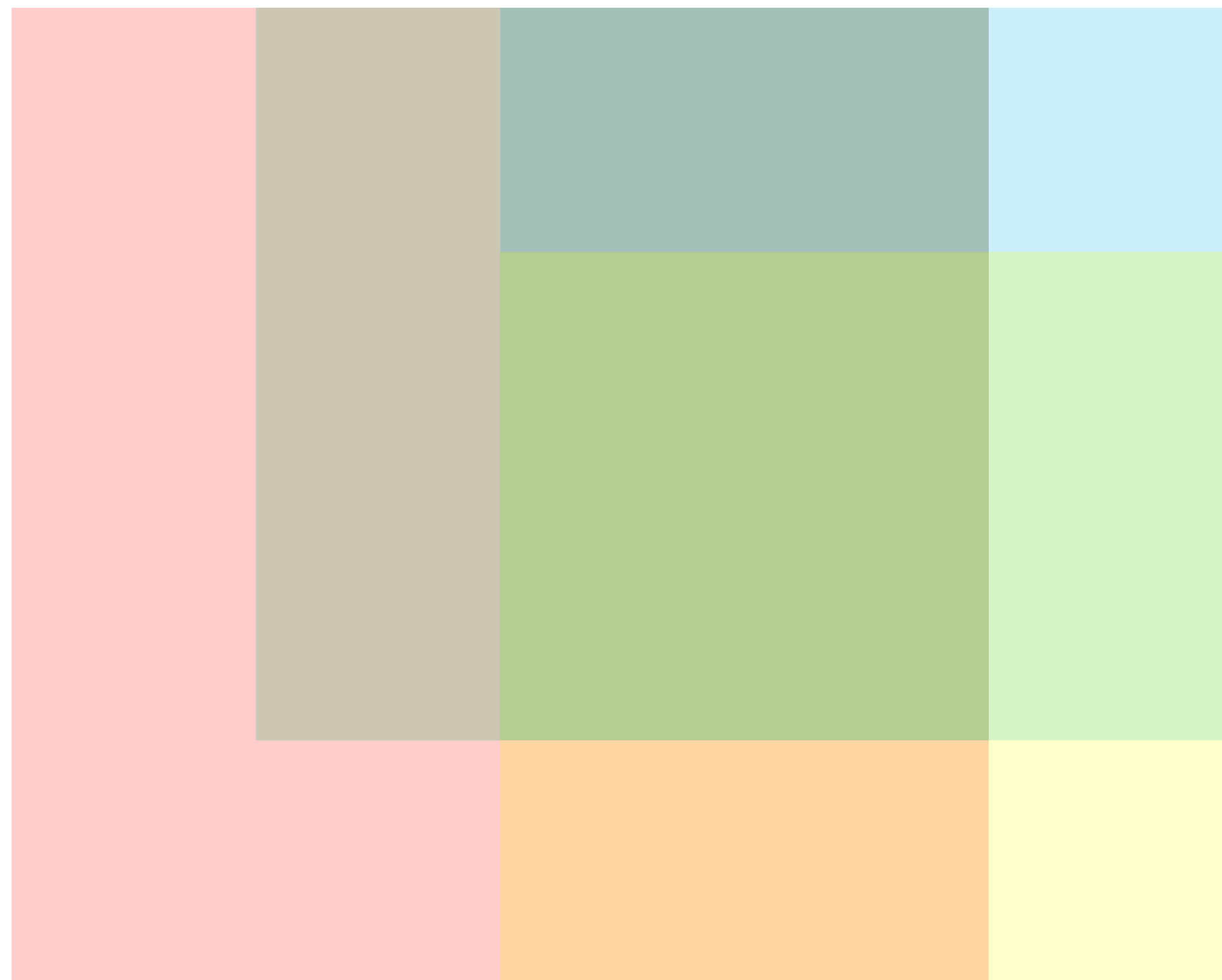
30

<https://www.acmicpc.net/problem/1915>



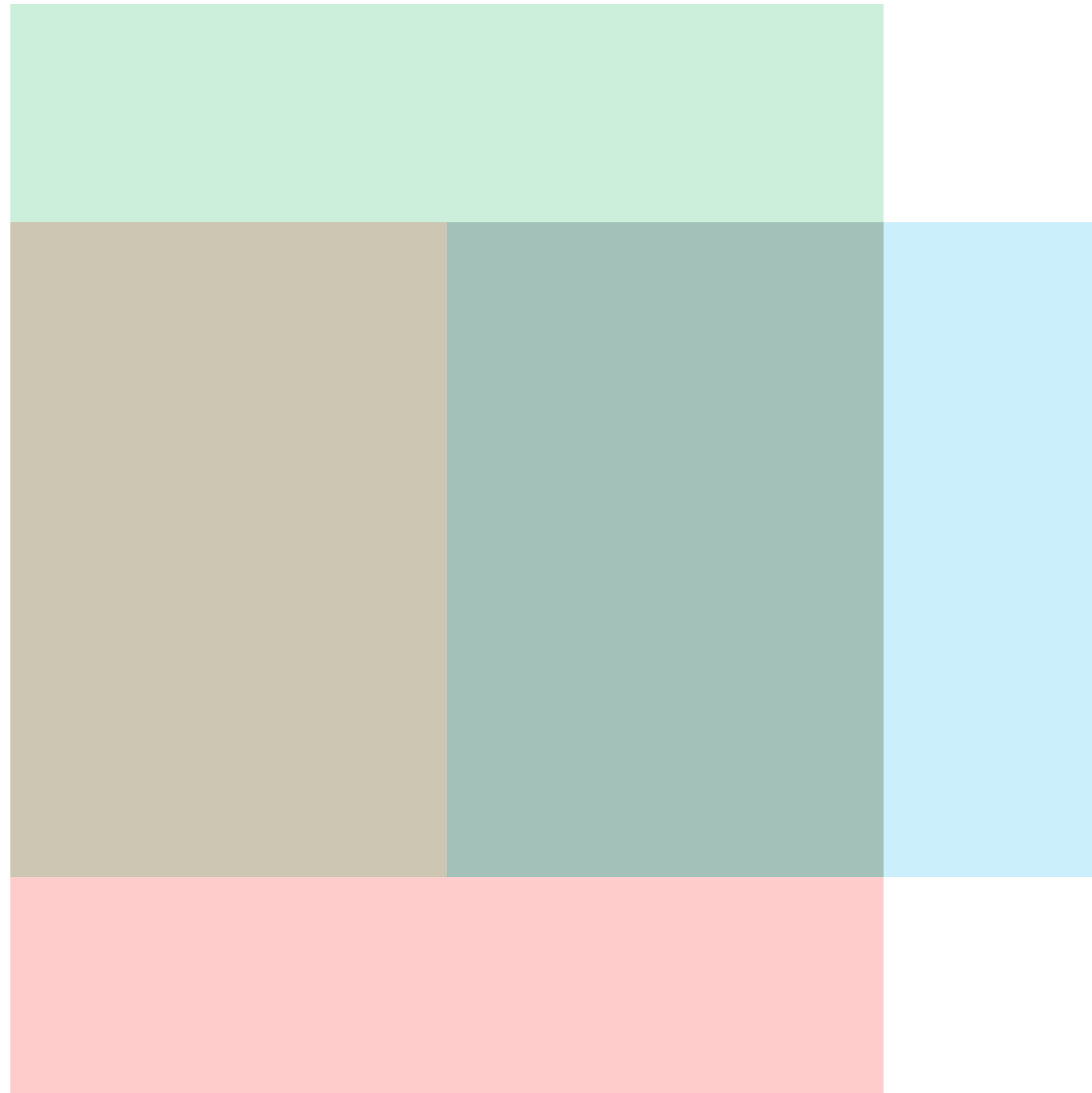
가장 큰 정사각형

<https://www.acmicpc.net/problem/1915>



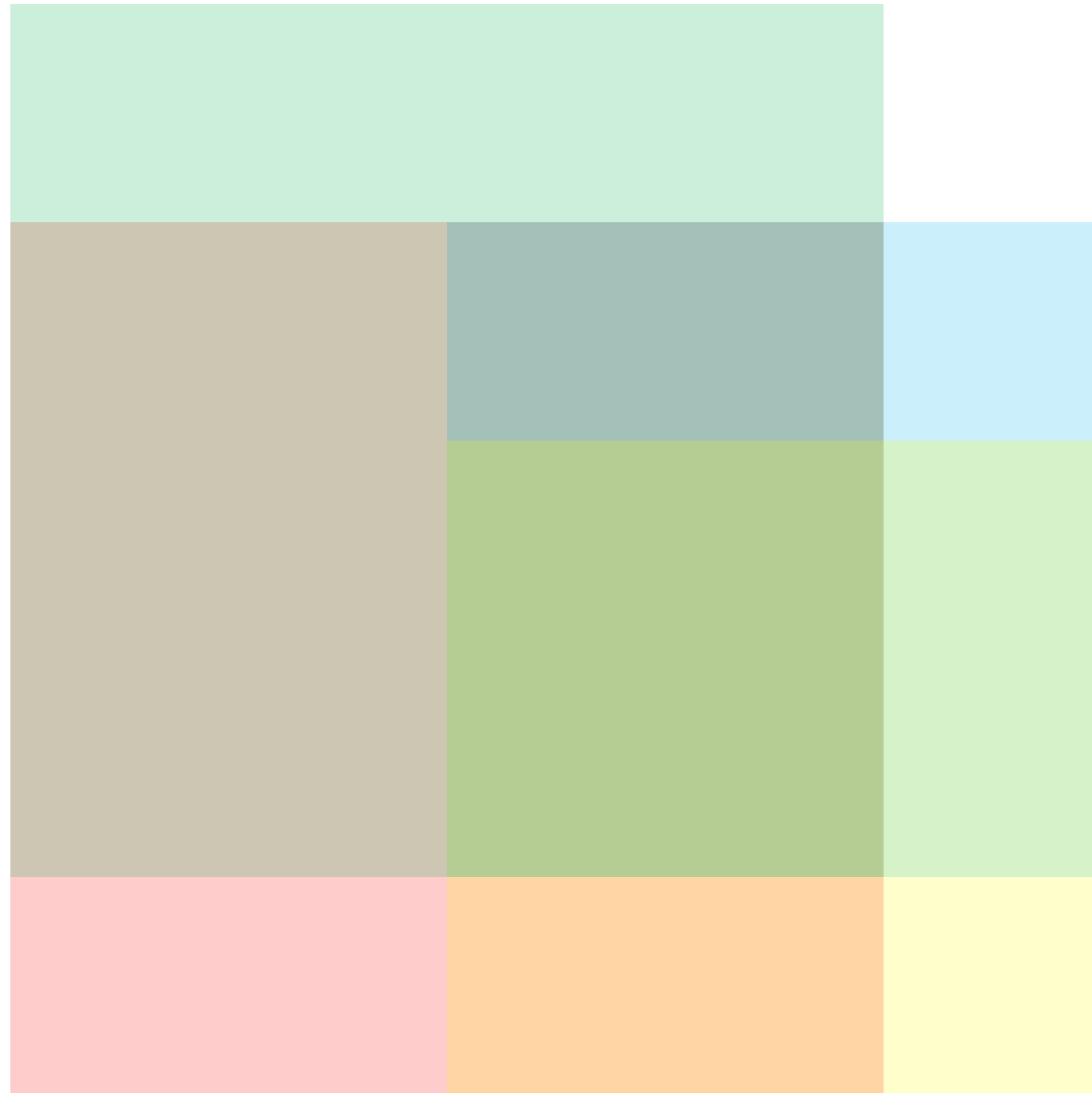
가장 큰 정사각형

<https://www.acmicpc.net/problem/1915>



가장 큰 정사각형

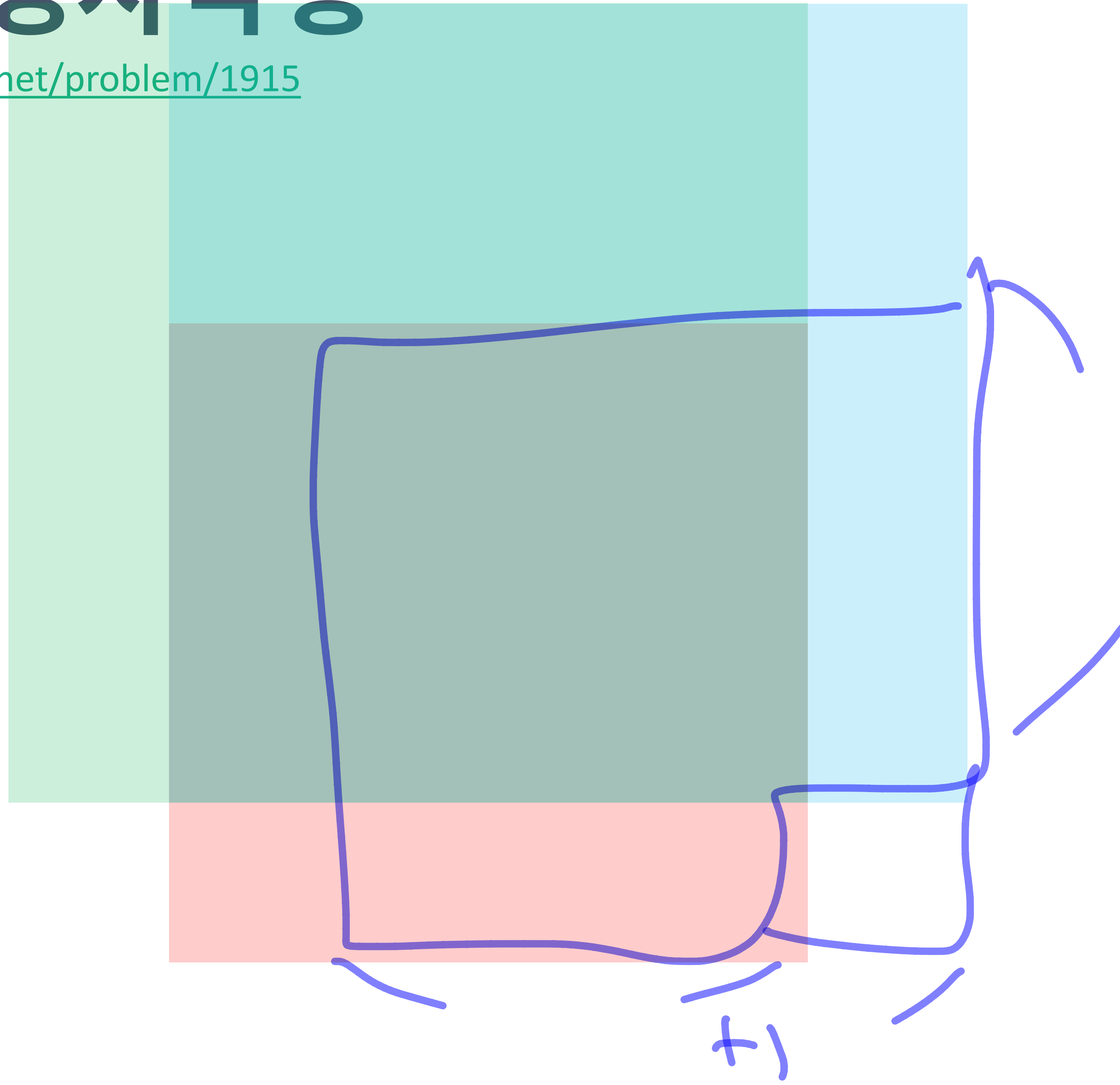
<https://www.acmicpc.net/problem/1915>



가장 큰 정사각형

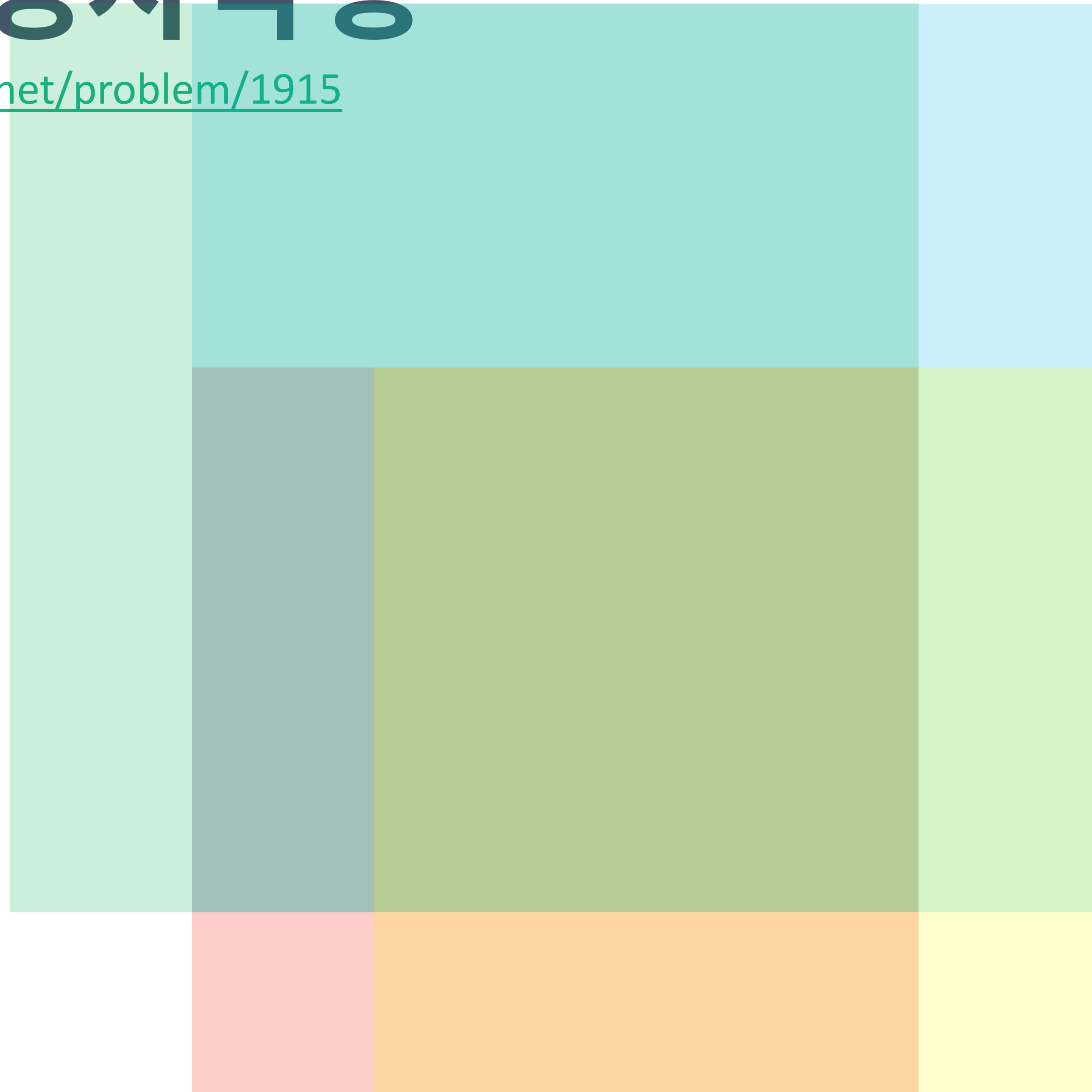
<https://www.acmicpc.net/problem/1915>

34



가장 큰 정사각형

<https://www.acmicpc.net/problem/1915>



가장 큰 정사각형

36

<https://www.acmicpc.net/problem/1915>

• $D[i][j] = \min(\{D[i-1][j-1], D[i-1][j], D[i][j-1]\}) + 1;$

← ↑ ←

가장 큰 정사각형

37

<https://www.acmicpc.net/problem/1915>

- 소스: <http://codeplus.codes/b1acef6090d8410abe688b55c3322053>

끝

코드 플러스

<https://code.plus>

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 codeplus@startlink.io 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.